

PHASE I FINAL REPORT

The Fault Monitoring & Diagnosis
Knowledge-based System for
Space Power Systems: AMPERES
(NAG8-721)

Submitted to

NASA/MSFC
MSFC, AL 35812

November, 1989

Submitted by

S.C. Lee
The University of Tennessee
Space Institute

(NASA-CR-185486) THE FAULT MONITORING AND
DIAGNOSIS KNOWLEDGE-BASED SYSTEM FOR SPACE
POWER SYSTEMS: AMPERES, PHASE 1 Final Report
(Tennessee Univ. Space Inst.) 44 p CSCL 10B

N90-12654

Unclass
0240376

63/20

Table of Contents

	Page No.
Executive Summary	1
1. Introduction	3
2. Technical Approach	12
3. Experimental Set-up	28
4. Tasks Completed in Phase I	32
5. Project Schedule and Tasks Planned in Phase II	36
6. Summary & Conclusions to Date	38
7. References	39
8. Appendix	40
i. Project Schedule	
ii. Publications	

Executive Summary

The objective of this project is to develop a real time fault monitoring and diagnosis knowledge-based system (KBS) for space power systems which can save costly operational manpower and can achieve more reliable space power system operation. The proposed KBS has been developed using the AMPS (Autonomously Managed Power System) test facility currently installed at NASA Marshall Space Flight Center (MSFC), but the basic approach taken for this project could be applicable for other space power systems. The proposed KBS is entitled "AMPERES" (Autonomously Managed Power-System Extendible Real-time Expert System).

This project is being carried out in two phases. Phase I was completed as of September 30, 1989, and currently Phase II is being performed. In Phase I, emphasis was put on the design of the overall KBS, the identification of the basic research required, the initial performance of the research, and the development of a prototype KBS. In Phase II, emphasis is put on the completion of the research initiated in Phase I, and the enhancement of the prototype KBS developed in Phase I. This enhancement is intended to achieve a working real time KBS incorporated with the NASA space power system test facilities. Three major research areas have been identified and progress has been made in each area. These areas are: (1) real time data acquisition and its supporting data structure; (2) sensor value validations; (3) development of inference scheme for effective fault monitoring and diagnosis, and its supporting knowledge representation scheme.

Currently, AMPERES is able to collect the real time operational data and to assess the power system operating status. The operational data including the fault data is generated using a data simulation program running on a separate computer and is transferred through the Ethernet to the host computer, Sun 386i, at the UTSI. Part of the operational data are collected from the actual ammeter and voltmeter measurements and the position sensor values installed in the fault injection and load simulation device (FILSD). A resistive load can be connected to the FILSD and can replace any one of the code generated loads by the data simulation program to create the operational environment close to the AMPS test facility at NASA/MSFC as possible.

Various faults and disturbances, such as overload, ground fault, battery cell open, solar array system failure, load connect and disconnect, etc., can be generated using the data simulation program incorporated with the FILSD. AMPERES is able to detect those faults or disturbances and to provide report to the operators, which includes fault kind, location, detecting sensors, severity, time, etc.

1. Introduction

1.1 Problem Statement

Electric power is a precious resource in space due to its extreme usefulness and strictly limited availability. Not only is it crucial to the operation of the crew members' life-support system, but also it directly affects the overall performance of a specific mission. For this reason, electric power must be available 24 hours a day throughout the mission period and must be properly managed to meet the power demand with high quality for the successful achievement of the mission objective.

The following are some of the specific problems which create difficulties in achieving highly reliable operation of the space power systems.

i. Expensive crew members' manpower:

Space power systems must be continuously monitored throughout the mission period. Maintaining this manpower in space, or even on ground as an alternative, is very expensive.

ii. Difficulties in accumulating the power system operation experience and expertise in space:

Necessity of the periodical crew members' rotation from space duty may make it difficult for crew members to accumulate enough experience and expertise in space power system operation, especially for an emergency or an abnormal operating state.

iii. Possibility of misoperation:

If a fault occurs, normally several alarms come up simultaneously because of the cascading effect of the fault. Also the remedial actions should be taken within appropriate time. Numerous incoming alarms together with time pressure for remedial action often create confusion even for the skilled operators, which may induce misoperation and further aggravate the operating state.

1.2 Objective of the Project

The objective of this project is to develop a real time fault monitoring and diagnosis KBS for space power systems which can solve or alleviate the problems stated above and can achieve more reliable space power system operation. The research work necessary to solve various inherent problems associated with the real time expert systems has also been performed in this project.

The following are the goals of the AMPERES:

- i. To relieve crew members or supporting staffs on ground from the power system monitoring tasks.
- ii. To determine the cause of any fault or disturbance, provide an explanation for the fault, determine the current status of the power system, and provide recommendations for remedial actions within appropriate time.
- iii. To perform sensor value validations to provide accurate operating state information to the fault monitoring and diagnosis KBS and the power system operator.
- iv. To carry out mid term and long term observations of the major operational parameters to identify the failure modes and compute the life expectancy of the major components such as battery systems, solar array systems, etc.

1.3 AMPS Test Facility

The AMPERES has been developed using AMPS test facility installed at NASA/MSFC [Fig. 1]. Major features of the AMPS are:

(1) a programmable solar array simulator which supplies 220 ± 20 Vdc directly to three power channels with a maximum power output of 75 kW; (2) an energy storage simulator which consists of a battery with 168 commercial nickel-cadmium (Ni-Cd) cells serially connected to provide a nominal dc voltage of 220 Volts and a capacity of 189 Ampere-hours; and (3) a load simulator which consists of nine resistive loads and one dynamic load that consume a total of 24 kW of power when operated at 200 Vdc. In addition, three Motorola 68000 microcomputer based controllers provide data retrieval and low-level decision-making for the power system.

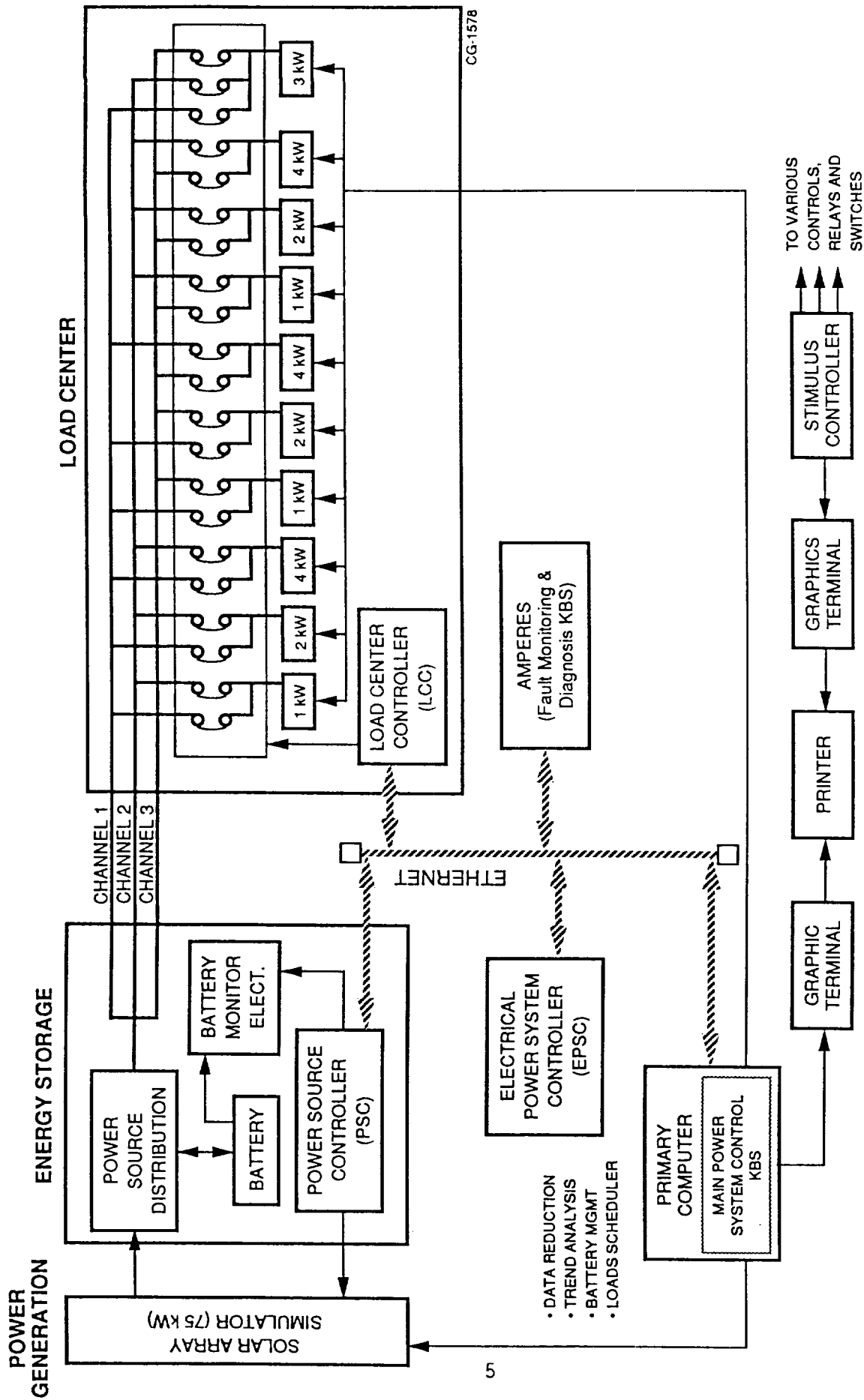


Figure 1. Autonomously Managed Power System (AMPS)

Detailed structural and functional description of the AMPS test facility is well documented in TRW final report [1]. D. Weeks discussed knowledge-based system (KBS) approaches employed in various electrical power system breadboards at NASA/MSFC including the AMPS test facilities [2,3]. L. Lollar described about the KBS development for automated load management for space power systems [4]. B. Walls developed a flexible prototype fault detection and recovery system concentrating on the load control center for AMPS called "Starr" using Intellicorp's Knowledge Engineering Environment (KEE) [5].

1.4 Past Work in Fault Monitoring and Diagnosis

A fault monitoring and diagnosis knowledge based system should be able to collect the real time operational data continuously and to assess the current power system operating status. If there exists any indication of a fault, it should be able to discern the actual occurrence of the fault from various transient status or noisy environment, and to find out the cause and the consequence of the fault within appropriate time. This real time operational constraints poses many complex and dynamic problems which are in the research state. These include the requirements of continuous expert system operation, handling of asynchronous events, temporal reasoning, nonmonotonic reasoning, response time, handling of transient state, and filtering of sensor noises and errors, etc. Detailed discussion of these problems and other relevant issues associated with the real time expert systems appear in [6-10]. Survey of current efforts and existing real time expert system tools and applications are documented in [11]. Most of the applications surveyed are in the prototype stage. There are few commercially available expert system shells for real time fault monitoring and control applications, such as Picon and G2 [6, 12], which are developed for various general system applications such as industrial process control and are not quite well suited for our specific applications.

The conventional expert system approach for fault monitoring and diagnosis of a physical system is performed normally by looking at the snapshot picture of the system state. Then appropriate fault patterns are generated by mapping the present numerical sensor values into a couple of descriptive terms such as "high," "normal," "low," etc. Then finally by comparing the operational state pattern thus

generated with the fault data patterns stored in the knowledge-base, matching fault patterns are identified. The major draw back of this approach is that it can handle only those faults whose patterns are explicitly stored in the knowledge base. For a physical system with reasonable complexity, enumerating all the possible fault cases is often very difficult and time consuming. Another draw back with this pattern matching approach is the mapping boundary problems. If a couple of sensor values in two similar operating states were very close but happen to be landed on both sides of the mapping boundaries of the descriptive terms, the resulting patterns become quite different. Furthermore, with the unavoidable sensor noises and errors in reality, the method is found to be almost impractical for realistic applications. To compensate these drawbacks and to provide a more generic and domain independent fault monitoring and diagnosis system, many researchers propose a causal model based reasoning approach which concentrates more on the designed function and behavior of each component in as physical system [13-15]. This idea seems ideal but a component may exhibit different behavior depending on its physical and functional environment and the combined effects of various components are often hard to predict by simply looking at each component's characteristics. Research emphasis in AMPERES have been put mainly in solving above stated problems, and the concentration areas are mentioned earlier.

1.5 Approaches Taken for the AMPERES

To perform several concurrent tasks, the main program in the AMPERES creates several processes upon initialization [Fig. 2]. The concurrent tasks required are the data acquisition task, user interface task and the main fault monitoring and diagnosis task. Interprocess communication is managed through shared memory.

Sensor data is categorized as critical or non-critical data based on the possible changing speed of the measurement values and the time resolution requirements for diagnosis. Several data buffers are created to provide back-up buffers in case of a fault leaving the present data and some of the pre and the post fault data intact for diagnosis. Detailed data structures and buffer operations for data acquisition is described in the next section.

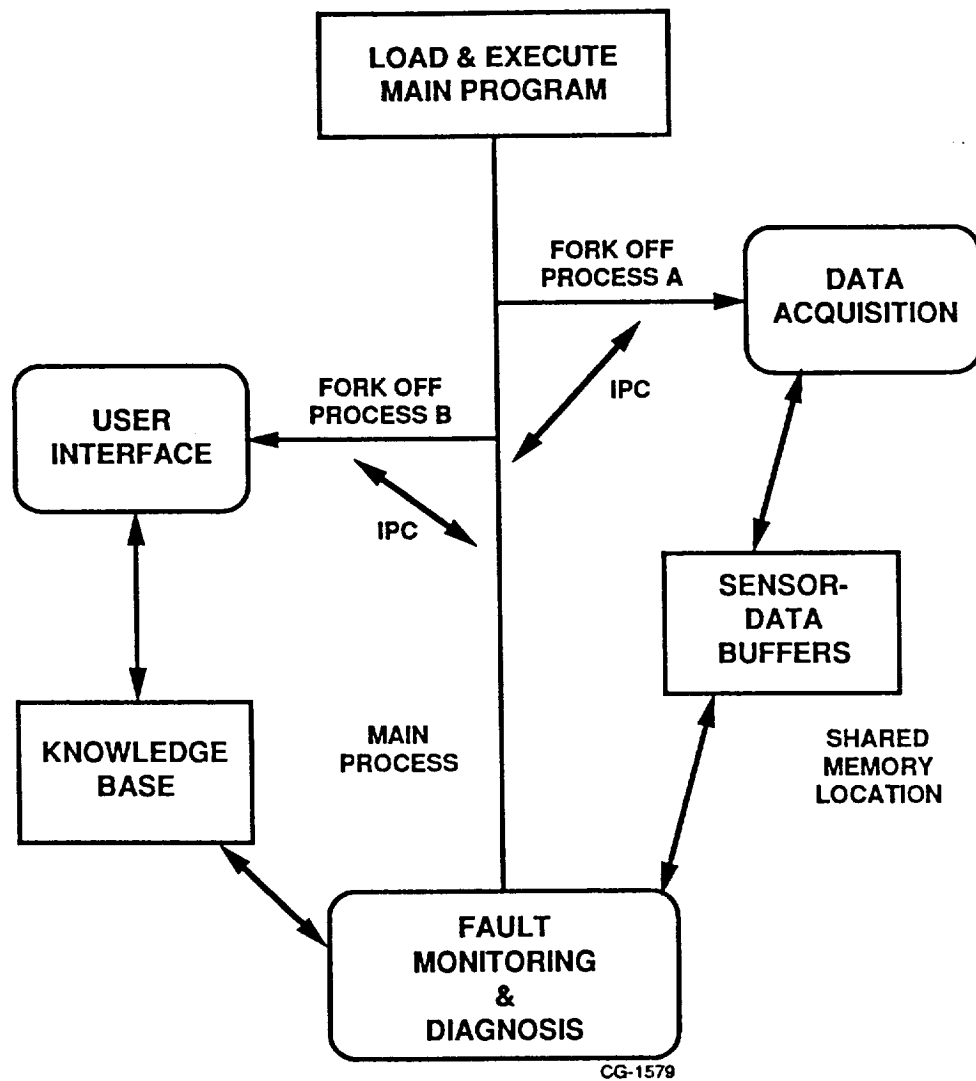


Figure 2. Concurrent Process Management in AMPERES

To generate a realistic measurement data and to inject faults safely into the AMPS, a fault injection and load simulation device (FILSD) is designed and assembled [Fig. 3]. In the load simulation mode, two resistive loads can be connected to the FILSD using batteries as a power source. The voltage and the currents across the loads are measured through signal conditioning circuits and the data acquisition board to create the operational environment close to AMPS test facility as possible. A time delayed fault can also be injected under computer control in the fault injection mode. A flexible data simulation program running on a separate computer from the host generates the power system operational data. It can replace any one of ten simulated load data with the actual incoming data from the FILSD.

Because of the significance of having valid data before any diagnosis process, a sensor data validation method based on the causal relations existing among sensors are currently being developed. The method utilizes the "Functional Environments" of sensors formulated based on the causal relations existing among sensor values. Then the sensor validation procedure propagates through the logical chains provided by these Functional Environments.

Each component in AMPERES, including sensors, is represented as an object. Each component representation includes the information about the component itself, its functional or logical environment, and its physical environment. The physical environment includes the information about those components which are directly connected to the current component, and the logical environment of a component includes the information about those components which are functionally or logically related to the current component. Starting from any sensor showing the indication of a fault, fault diagnosis is performed by tracing down or propagating through the sensor's logical environment with expectation and by providing justifications of or reasoning about each sensor value along the logical path.

The knowledge base includes the generic method descriptions for fault diagnoses instead of enumerating all the possible specific fault cases of the faulty data patterns. This keeps the size of the knowledge base small and coherent. Each fault possibility is diagnosed by a specialized knowledge group associated with each fault kind. Each knowledge group is composed of generic rules which can either assert the

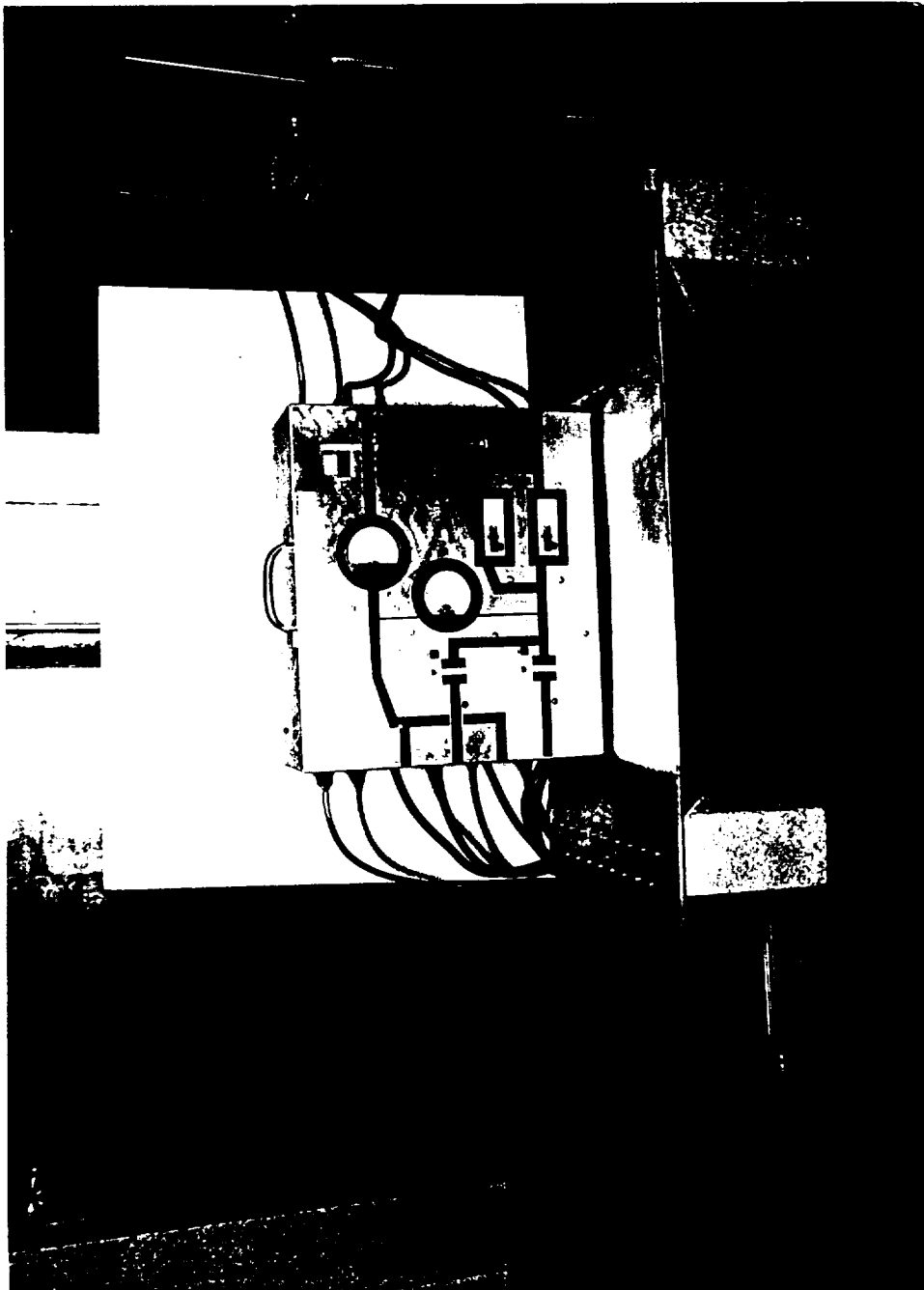


Figure 3. Fault Injection & Load Simulation Device

associated fault or freely invoke other knowledge groups for other fault possibilities. After the assertion of a fault, a rule can further probe different abstraction level presentation of the fault if necessary. Before a knowledge group is invoked, a context is set around a sensor to pass a default sensor associated with that knowledge group. This context switching together with the development of semantic primitives enable the rule representation more natural and clear. Combined with the “component centered” approach described in the previous paragraph, this inference and knowledge representation scheme provides a powerful reasoning tool for fault monitoring and diagnosis tasks.

2. Technical Approach

2.1 System Architecture

AMPERES is composed of five major functional models to efficiently perform the required tasks [Fig. 4]. The fault monitoring and diagnosis task is decomposed into several subtasks and each subtask is performed by a specialized module. In Phase I, program codes implementing each of the functional modules has been initiated except the Natural Language Interface, the load of Load Schedule KB, and the Statistic KB. The functions of each module are as follows:

(1) Main Controller

The Main Controller is a task oriented inference engine which is organized and tuned to perform the given fault monitoring and diagnosis tasks. It decides the current task of the AMPERES by invoking appropriate modules based on inputs from the sensors and other KBS modules. The Main Controller also includes a submodule, the Data Acquisition Module, except the Inference Module.

The Data Acquisition Module is in charge of reading in the incoming sensor values through the Ethernet and storing them into appropriate data buffers such that other AMPERES' modules can access the data.

(2) Status Monitor

The Status Monitor is in charge of assessing the current power system operational status. It is activated by the Main Controller after each sensor value scan cycle. It is composed of 3 submodules; the Expected State Generator, the Present State Confirmer, and the Status Evaluator.

The Expected State Generator is in charge of generating an expected normal operating state based on the current operational context. The expected state is filled in the appropriate attribute slots in the component representation. The expected state is then used as a reference state by the Status Evaluator in assessing the actual operating state.

The Present State Confirmer is in charge of formulating the actual current

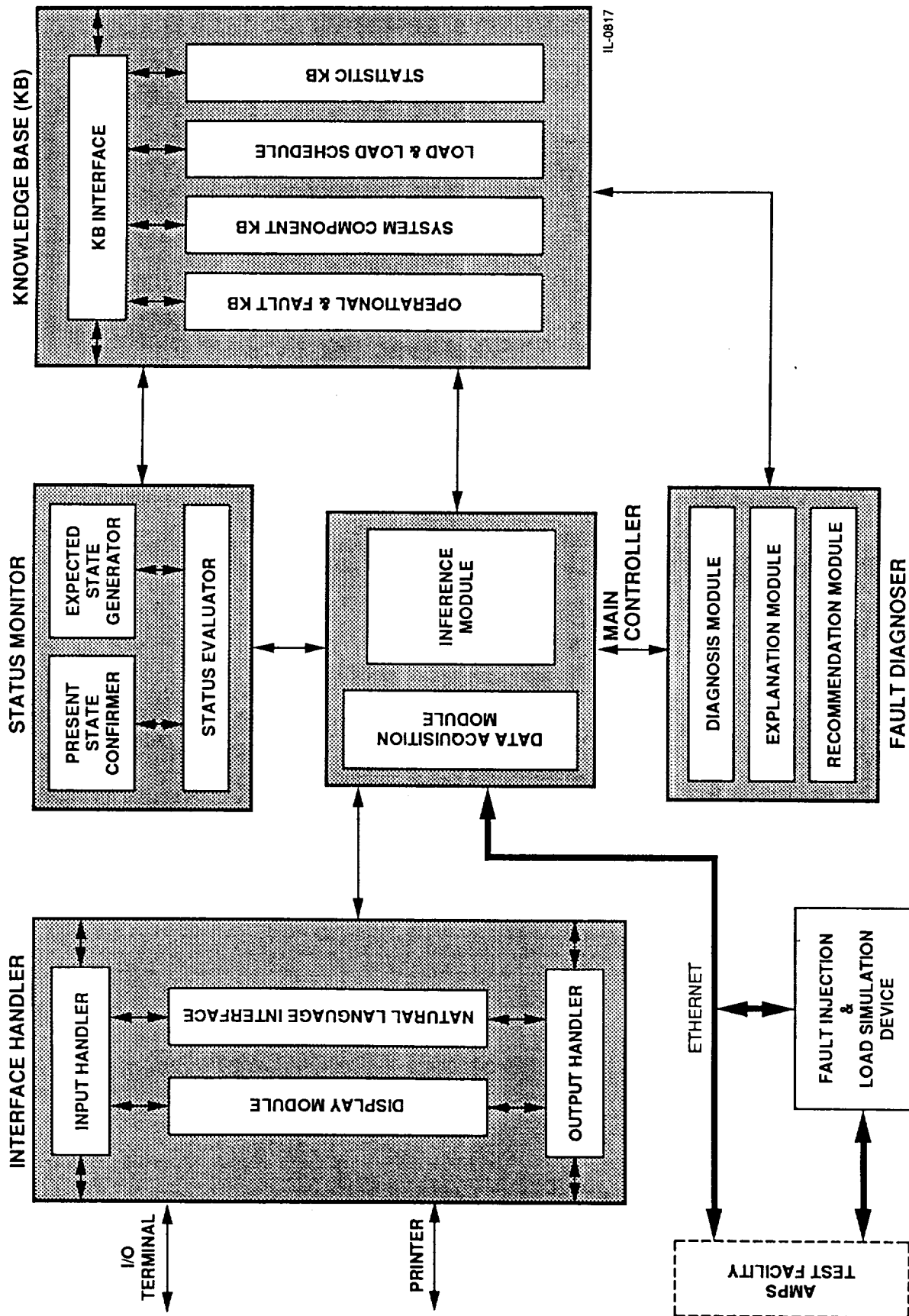


Figure 4. **AMPERES**
(Autonomously Managed Power-system Extensible Real-time Expert System)

operating state in the knowledge base from the various sensor values. The collected sensor data can be validated through a sensor value validation process to insure the correctness. Sensor failures can also be found out during this validation process. Then the sensor values are used to update the appropriate attribute slot values in the respective sensor representations.

The Status Evaluator is in charge of assessing the current operating status. It compares the two states obtained by the Expected State Generator and the Actual State Confirmer. If the current operating state is turned out to be a faulty state the Status Evaluator informs the Main Controller of it, which in turn invokes the Fault Diagnosis Module.

(3) Fault Diagnoser

Once a fault or a disturbance is identified by the Status Monitor, the Fault Diagnoser tries to find out the cause of the fault. It also gives recommendation for the necessary corrective actions to the power system operator. The Fault Diagnoser is composed of three submodules; the Diagnosis Module, the Explanation Module, and the Recommendation Module.

The Diagnosis Module tries to find out the cause of a fault and its consequences.

The Explanation Module provides explanation about the fault. It also answers the operator questions about the fault.

The Recommendation Module recommends the operator for the necessary corrective actions. The corrective actions are listed in the order of required action sequence.

(4) Knowledge Base (KB)

Required knowledge for performing the fault monitoring and diagnosis are organized and stored in appropriate forms in the KB for ease of manipulation and fast access by other major modules. The KB is composed of four submodules; the Operational & Fault KB, the Load & Load Schedule KB, the System Component KB, and the Statistic KB.

The operational KB includes the current operating status of the major system components including the sensor values. It also includes the normal expected behaviors of the power system components and the anticipated values from the sensors. The fault KB includes power system behavior during the faults, cascading effect of the faults and the associated sensor values, and the procedural knowledge required to filtering out the faulty components.

The load KB includes the information such as load size, and load characteristics. A load can be continuous, intermittent, or random. It also can be a pure resistive load of an inductive load requiring large start up inrush current. The load schedule KB includes the information about the load schedule enabling the AMPERES to have an anticipation on the scheduled change in power system operational status.

The System Component KB includes the information about the system components. The information about the system topology, both design and operational, is embedded in the representation of each component as a physical environment. Each component representation also includes the information about various logically or functionally related components to facilitate the fault monitoring and diagnosis task.

The Statistic KB includes the information about the fault statistics. It is used for the AMPERES in learning about the fault behavior and frequencies, and in updating the heuristic knowledge.

(5) Interface Handler

The Interface Handler is in charge of processing various I/O and is composed of four submodules; the Input Handler, the Output Handler, the Graphics Processor, and the Natural Language Interface. The function of these modules are self explanatory.

2.2 Data Acquisition System Design

Upon initialization of the AMPERES, the process running the main program forks off two processes, i.e., the data acquisition process and the user interface process. Then the three processes initialize their internal variables and necessary data

structures and run concurrently. Interprocess communication is managed through shared memory.

Sensor data is categorized as critical or non-critical data in AMPERES. The critical data set contains 58 analog values and 42 digital values and collected every 10 ~ 100 milliseconds. The non-critical data set contains 212 analog values and is collected every second. The collected critical and non-critical data are stored in separate circular buffers.

The data acquisition process creates three circular buffers, two for critical data sets and one for non-critical data sets, in shared memory locations. The buffers are utilized to store data from the sensors and supply them to the inference engine and the display software. The buffers of each critical and non-critical data set are referred to as the primary buffer and the secondary buffer, respectively.

Upon initialization, the data acquisition process starts filling up the primary buffers. Global pointers are maintained by the data acquisition process to make known to the fault monitoring and diagnosis process and the user interface process the latest available data. Each time a new data set is written to the buffers, global pointers are also updated to point to the latest data set [Fig. 5]. The fault monitoring and diagnosis process accesses the latest data and performs the assessment of the power system state. If the fault monitoring and diagnosis process finds out any indication of an abnormality in the system's operational state, it informs the data acquisition process of the fact such that the data acquisition process can perform the buffer switching operation.

A system status flag is also created in the shared memory location by the fault monitoring and diagnosis process to facilitate the buffer switching operation. The status flag has 4 state values [Fig. 6]. Initially, state 1 is set by the data acquisition process when it starts filling the data buffers. Then the data acquisition process checks this status flag each time before it fills the buffer with the collected data set. If the status flag value is 1, then the data acquisition process continuously fills the primary buffers. If the fault monitoring and diagnosis process detects any abnormality in the power system's operational state, it sets the status flag value

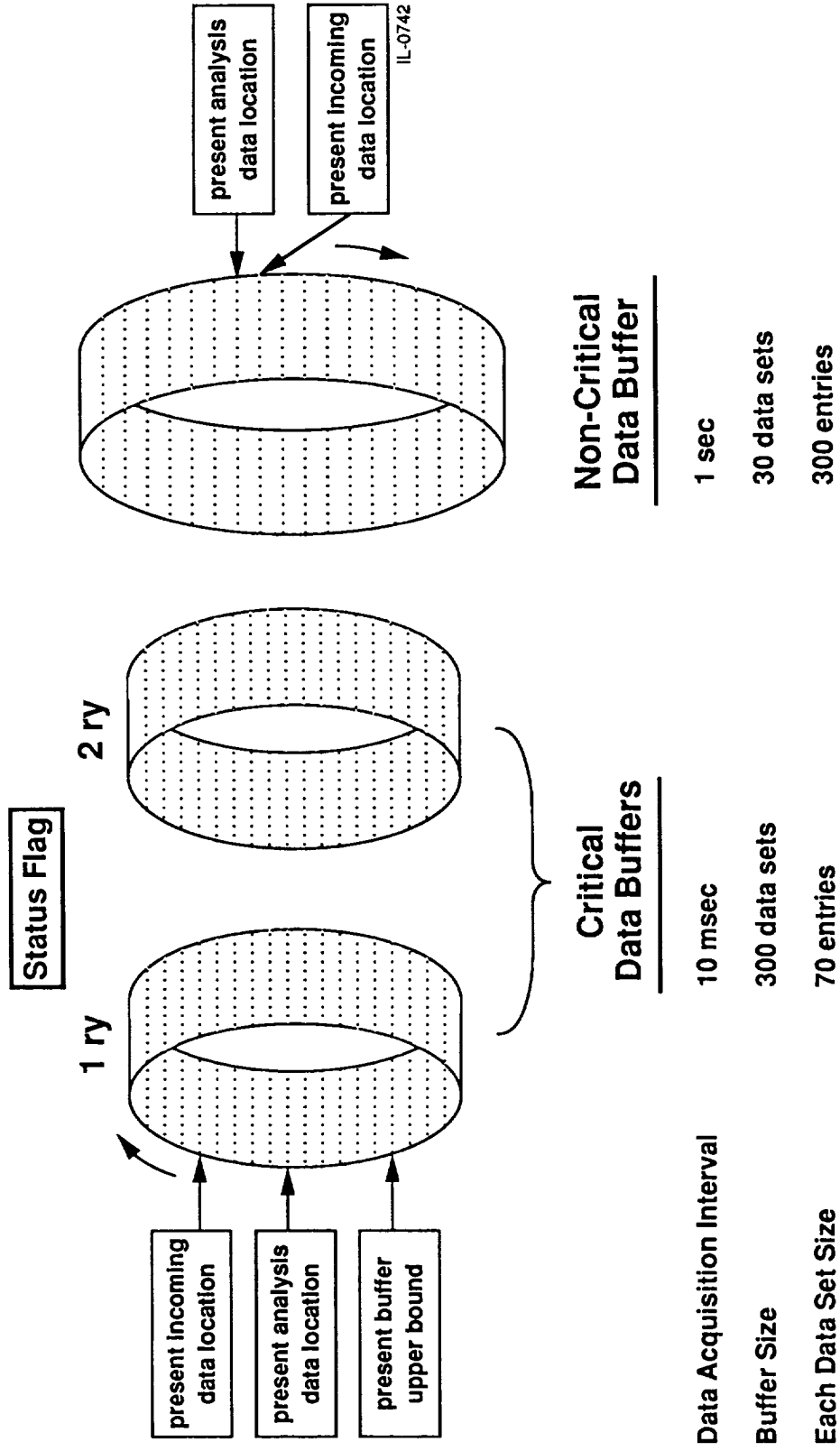


Figure 5. Data Structures for Data Acquisition

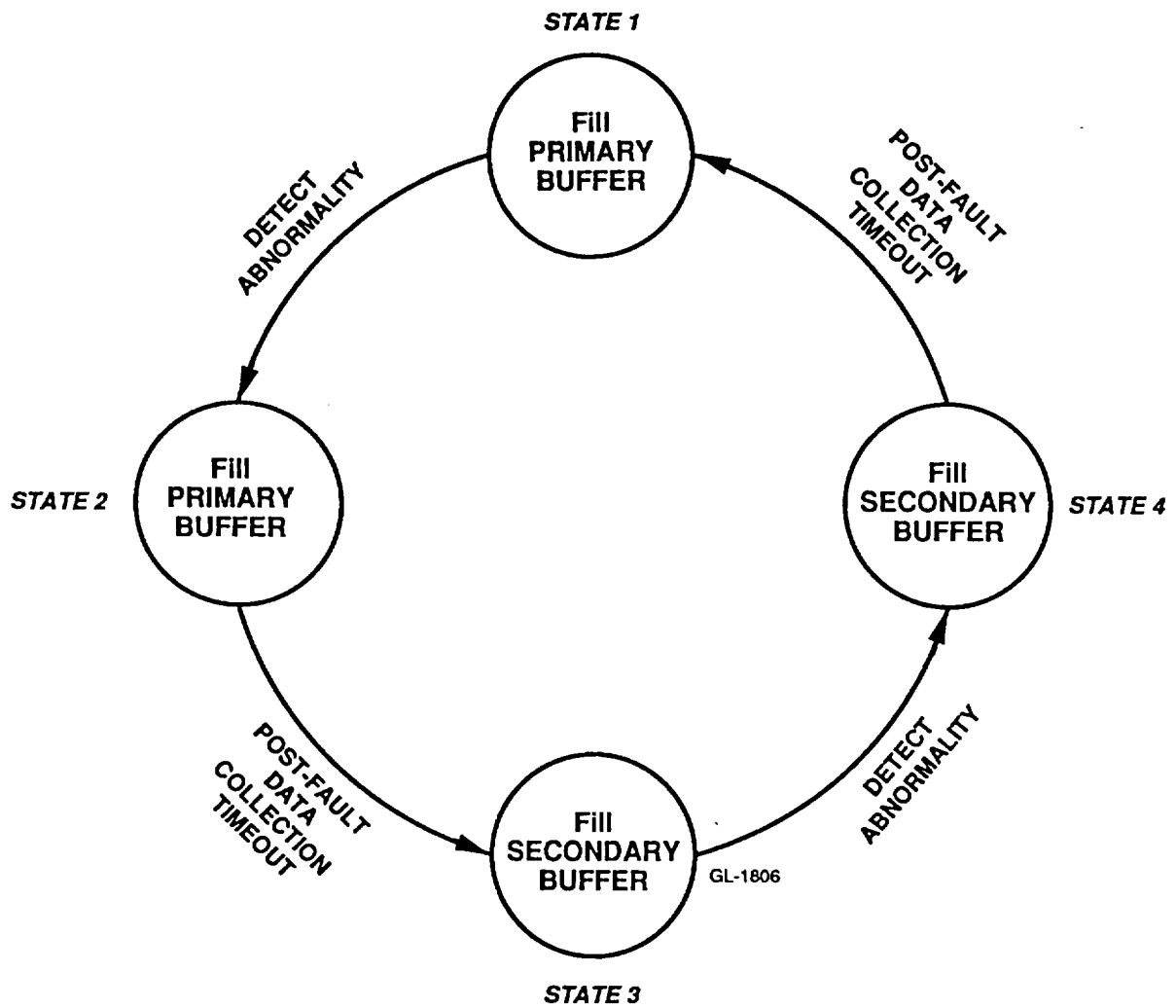


Figure 6. **Critical Data Acquisition Process
State Transition Diagram**

as 2. Upon noticing the flag value set to 2, the data acquisition process fills the primary buffers for about 1 more second, sets the status flag value as 3, and jumps to the secondary buffers, leaving the primary buffers intact until the diagnosis for the abnormality is completed by the fault monitoring and diagnosis process. Since the buffers, including the data showing the abnormality remains unchanged and the post fault data is also available, the fault monitoring process has freedom of accessing arbitrary pre and post fault data necessary for diagnosis. Upon completion of the diagnosis with the current data, the fault monitoring and diagnosis process resumes its normal operational state assessment task and will begin accessing the latest data in the secondary buffers marked by the global pointers. Again, if any abnormality is found, the status flag is set to 4 by the fault monitoring and diagnosis process. Upon detecting state 4, the data acquisition process fills in the secondary buffers for another second, resets the status flag as 1, and jumps back to the primary buffer.

2.3 Fault Injection and Operational Data Simulation

A Fault Injection and Load Simulation Device (FILSD) has been designed and assembled, which will be used in injecting various types of faults safely into the actual AMPS test facility installed at NASA/MSFC to obtain actual fault data [Fig. 3]. It will also be used in generating reduced scale operational and fault data using small size loads for extensive fault testing required to tune the AMPERES to the actual operating environment.

A data simulation program is written in C and is running on the PC to generate simulated operational data and transfer the data to the host computer through the Ethernet. Various faults and disturbances, such as overload, ground fault, battery cell open, solar array system failure, load connect and disconnect, etc., can be generated using the data simulation program incorporated with the FILSD. Any one of the ten AMPS' loads can be replaced with the actual load connected to the FILSD interactively and the actual current and the voltage measurements are sent to the Sun 386i for the replaced load. The orbital day and night period can also be set and the simulated operational data changes accordingly whenever day to night or night to day transition occurs.

2.4 Sensor Value Validation

In a fault monitoring and diagnosis process for a physical system whose operating state is monitored by numerous remote sensors, like AMPS test facility, one of the crucial steps involved is to validate the incoming sensor values. Trying to assess the system operating state based on false sensor values is not only futile but also is often detrimental to the monitored system and is normally accompanied with significant economical losses if any control action is taken based on such false sensor values. Yet in reality there are high chances that a sensor may give a false value, which can be either temporary or permanent. Research is currently being performed to lay a ground work for the sensor value validation procedure upon which the validation procedure for a specific system can be built systematically in corporation with the system's fault monitoring and diagnosis knowledge based system.

The method utilizes the "Functional Environment" (FE) of the sensors, which is the set of causal relations existing among sensors. For example, from the simplified one-line diagram of the AMPS [Fig. 1, 7], the FE of the feeder ammeter "AF1" can be formulated as follows:

$$\begin{aligned}
 R_1 : \quad AF1 &= (K1A)(AL1) + (K2A)(AL2) + (K3A)(AL3) + (K4A)(AL4) \\
 &\quad + (K5A)(AL5) + (K6B)(AL6) + (K10A)(AL10) \\
 R_2 : \quad AM &= AF1 + AF2 + AF3
 \end{aligned}$$

where

$R_1, R_2 = \text{Causal Relations}$

$KiA = \text{Magnetic switch position}$

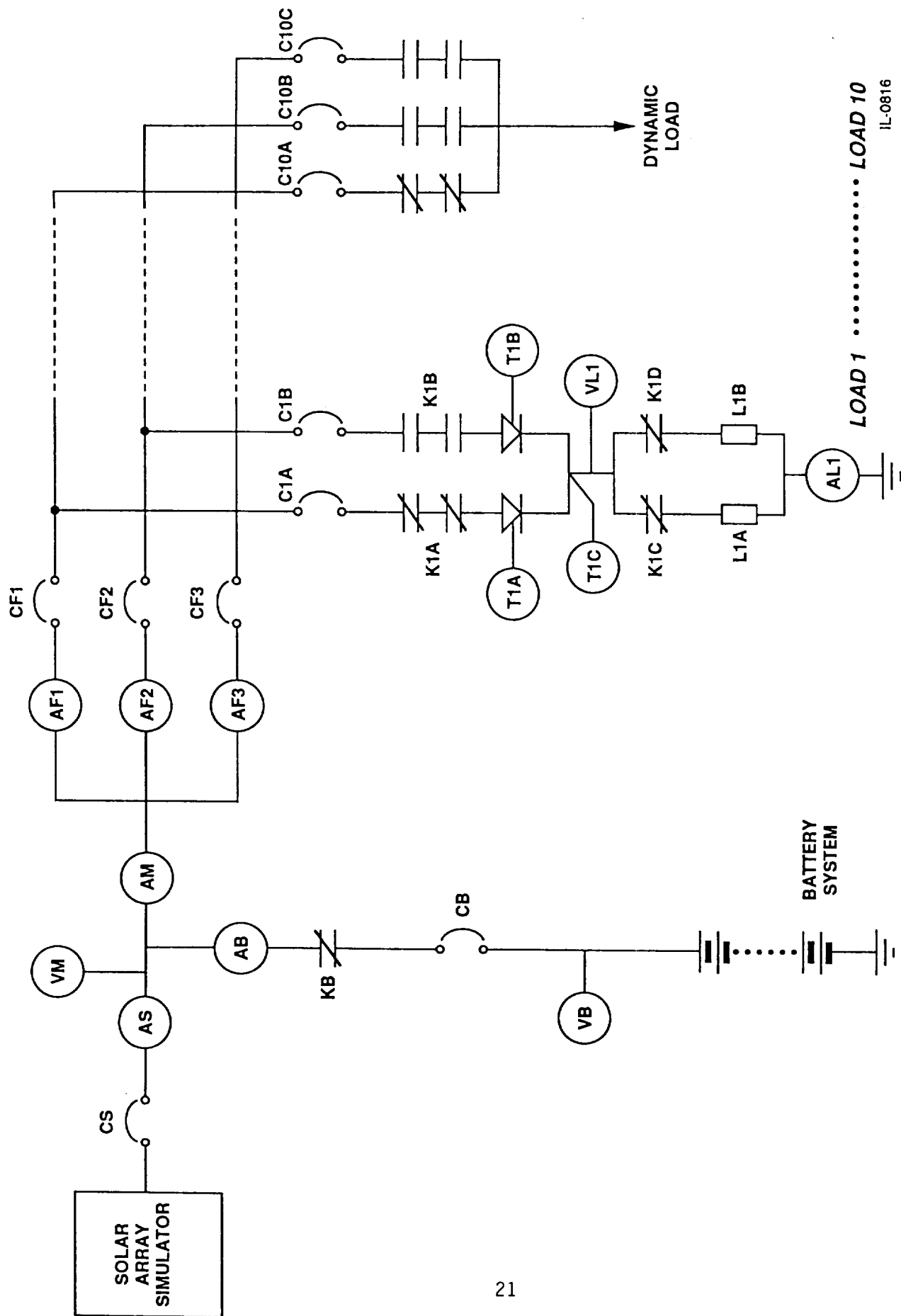
0 = open

1 = close

$i = 1, 2, 3, 4, 5, 6, 10$

$AF1, ALi = \text{Ammeter values}$

$i = 1, 2, 3, 4, 5, 6, 10$



IL-0816

Figure 7. Simplified One Line Diagram for AMPS

Then “Unit Functional Environments” (UFE’s) for ammeter AF1 can be formulated as a tuple such as

$$e_1 = (\{AF1, AL1, AL2, AL3, AL4, AL5, AL6, AL10\}, R_1)$$

$$e_2 = (\{AM, AF1, AF2, AF3\}, R_2)$$

Finally the FE of the AF1, \mathcal{E}_1 , becomes

$$\mathcal{E}_1 = \{e_1, e_2\}$$

The AF1 value can be validated if one of the relations R_i of $e_i \in \mathcal{E}_1$ is consistent. If none of the R_i is consistent AF1 value is invalidate.

Above validation procedure is based on the assumption that no two sensors in a FE of a sensor can have errors exactly in the same data scanning interval. This assumption is made based on the following facts:

- i. The data scanning interval, i.e., the time between the two consecutive data scans, is short, about 10 ~ 100 ms in AMPERES.
- ii. The probability of two independent events occurring at the same time in the continuous distribution is zero.
- iii. The number of sensors in a FE of a sensor is small, normally are less than ten.

Above assumptions may be released for a specific sensor if there exists a good chance that two sensors in the same FE of that specific sensor may fail exactly at the same data scanning interval. Some of the sensors may have only one UFE, subsequently having only one R_i , in its FE. In such case, if the only R_i is not consistent, the validation procedure may be applied to each sensor member in the UFE recursively to check if all other sensor values in the UFE can be validated. Detailed validation procedure will appear in future publications.

2.5 Knowledge Representation

As mentioned earlier, each component in the AMPS is represented as an object in AMPERES using the structure definition in the Common Lisp. The example of

an ammeter representation is shown in Fig. 8. The information included in each component representation can be categorized in the following three groups:

i. The information about the component itself.

The information about the component itself are normally the design data, the expected operational value, the present operational value, graphics information for display, etc.

For example, from Fig. 8, component-id, one-of, present-expected-range, normal-expected-range, present-value, trend, faulty-state, and graphics are the slots representing the information about an ammeter.

ii. The Functional Environment of the component.

The functional or logical environment of a component includes the information about those components which are functionally or logically related to the current component. This information is essential in collecting the supporting evidences and checking the cascading effects of a fault.

From Fig. 8 assoc-cb, i.e. functionally associated circuit breaker, assoc-ms, connect-load, parent-ammeter, children-ammeter, assoc-voltmeter are the slots related to the functional environment of an ammeter.

iii. The Physical Environment of the component

The physical environment includes the information about those components which are physically connected or attached to the current component. This information is necessary in identifying the extension of a faulty location and for graphical display of the system. From Fig. 8, connect-terminal and location slots are examples of such information.

Fault monitoring and diagnosis knowledge is implemented in production rule forms. Fig. 9 shows an example of a battery system failure rules. Rule languages are defined to write a rule close to natural language form as possible. For simplicity in the rule expression, "If clauses" are implicitly "AND ed." A rule can refer to any other knowledge, which is represented as a group of rules.

2.6 Fault Monitoring and Diagnosis Scheme Development

The fault monitoring process starts from checking the circular data buffers

Component Representation and Its Environment

```
----- Ammeter Frame -----  
(defstruct ammeter  
  (component_id)  
  (connect_terminal )  
  (location)  
  (one_of 'load_ammeter)  
  (assoc_cb )  
  (assoc_ms )  
  (connect_load )  
  (parent_ammeter )  
  (children_ammeter )  
  (assoc_voltmeter)  
  (present_expected_range )  
  (normal_expected_range )  
  (present_value)  
  (trend 'steady)  
  (faulty_state 'no)  
  (signal_ckt )  
  (graphics ))
```

Figure 8. Representation for an Ammeter

Rule Representation and Rule Languages

```
; battery_system_failure_rules.lisp
;
; (setq battery_system_failure_rules '(
      (rule1
        (if (changed present_value from normal to 0) )
            (changed battery_ammeter from normal to 0))
      ;
      (then
        (assert present_sensor battery_system_failure)
        (check battery_voltmeter battery_open)
        (check battery_voltmeter battery_ground)
        (check battery_voltmeter battery_short)))
```

Figure 9. Battery System Failure Rule Example

whether new data is available since the last access to the buffers. This is done by comparing the present buffer location pointers, which are updated by the data acquisition process each time new data is obtained, with the previous buffer pointers. If a new data is available, the data is read in from the buffer and the appropriate slot values of each sensor frame is updated.

Then the expected operating state or the reference operating state is generated from the current operational context and from the system design information. The expected operating state is generated such that it can minimize the number of sensors deviating from their expected ranges as possible in case of faults or disturbances. For example, if a load is connected to a feeder and the corresponding switches are closed, then there should be certain load voltage range expected from the system design, and the expected load current can be computed from that expected voltage range and the load size. If suddenly the load current goes to zero because of no load voltage, the load current will be out of the normal expected range. But from the present operational context, the load current is naturally expected to be zero since there is no load voltage, and consequently the load current value of zero is considered to be within the normal range.

For the load voltage, if it goes to zero because of the no system voltage, then the load voltage of zero is also considered to be within its normal range. The only sensor value out of the expected range in this case is the main voltmeter value whose expected ranges are designed system nominal voltages.

The above approach significantly reduces the burden of the fault diagnosis process by enabling the fault diagnosis process to concentrate on examining the sensor values directly responsible for the faults.

Once an abnormality is found, the fault monitoring process creates an abnormality list and passes the list to the fault diagnosis process. The abnormality list is a list of pairs, i.e. an association list, and each pair include the sensor name showing abnormality and the sensor kind it belongs to. For example, ((Voltmeter VM) (Ammeter AL1) . . .) is an abnormality list showing that the present values of the voltmeter VM and the Ammeter AL1 exceed their expected ranges respectively.

Upon receiving an abnormality list, the fault diagnosis process picks up one pair each time from the list and examines the situation by invoking the appropriate rule groups. Rules are grouped such that each rule group is specialized in resolving a specific situation or a fault. The first rule group which is called every time a pair from the abnormality is picked up is the major sensor rule group, such as voltmeter rules, ammeter-rules, ckt-breaker-rules, etc. These rule group include rules which exhaustively categorize the present sensor values or value trends and invoke appropriate rule groups in sequence to check all the possibilities.

Whenever each rule group is invoked, a context is set around a sensor which is going to be the center of the universe in examining its logically related sensor values. This context switching enables the rule expression to be simple and natural.

Once a fault is found, a fault object is created, which includes the information related to the faults. Before the fault diagnosis process picks up the next pair in the abnormality list it deletes those pairs who are included in the abnormality list because of the cascading effects of the present fault found. The fault monitoring process repeats the above process until all the pairs in the abnormality are checked. When the diagnosed fault with the present pair is same with one of the faults found with the previous pairs, the result is ignored.

Finally all the faults found are passed to the interface process for display and the control is passed to the fault monitoring process to assess the system operating state with new data again.

3. Experimental Set-up

In order to facilitate load simulation and to create a similar data acquisition environment with the AMPS test facility at the UTSI, an IBM PC compatible computer was interfaced to the FILSD, outfitted with an Ethernet interface and connected to the Sun 386i system [Fig. 10]. The FILSD can be used in two modes, the Fault Injection Mode and the Load Simulation Mode. Two circuit breakers, 30A and 100A ratings respectively, are provided to test fault currents at different magnitudes and to provide adequate protection during the fault injection period. The line voltage is measured and conditioned to interface to the PC analog input hardware via a voltage divider on the signal conditioning board. Line current is detected by a 200 AMP/50mv shunt and amplified by the signal conditioning board to the proper level of the PC analog input hardware [Fig. 11, 12]. Control logic allows for manual or computer controlled load connection. Selecting manual begins a time delay (adjustable for .3 to 3 seconds) and signals the PC to start data acquisition. Upon timeout of the time delay relay, the associated contactor connects the load. Selecting computer control allows the computer to connect or disconnect the loads under software control.

The PC collects data from the FILSD much the same way the AMPS does. Raw data is converted to engineering unit data and buffered in the PC and transferred to the Sun 386i.

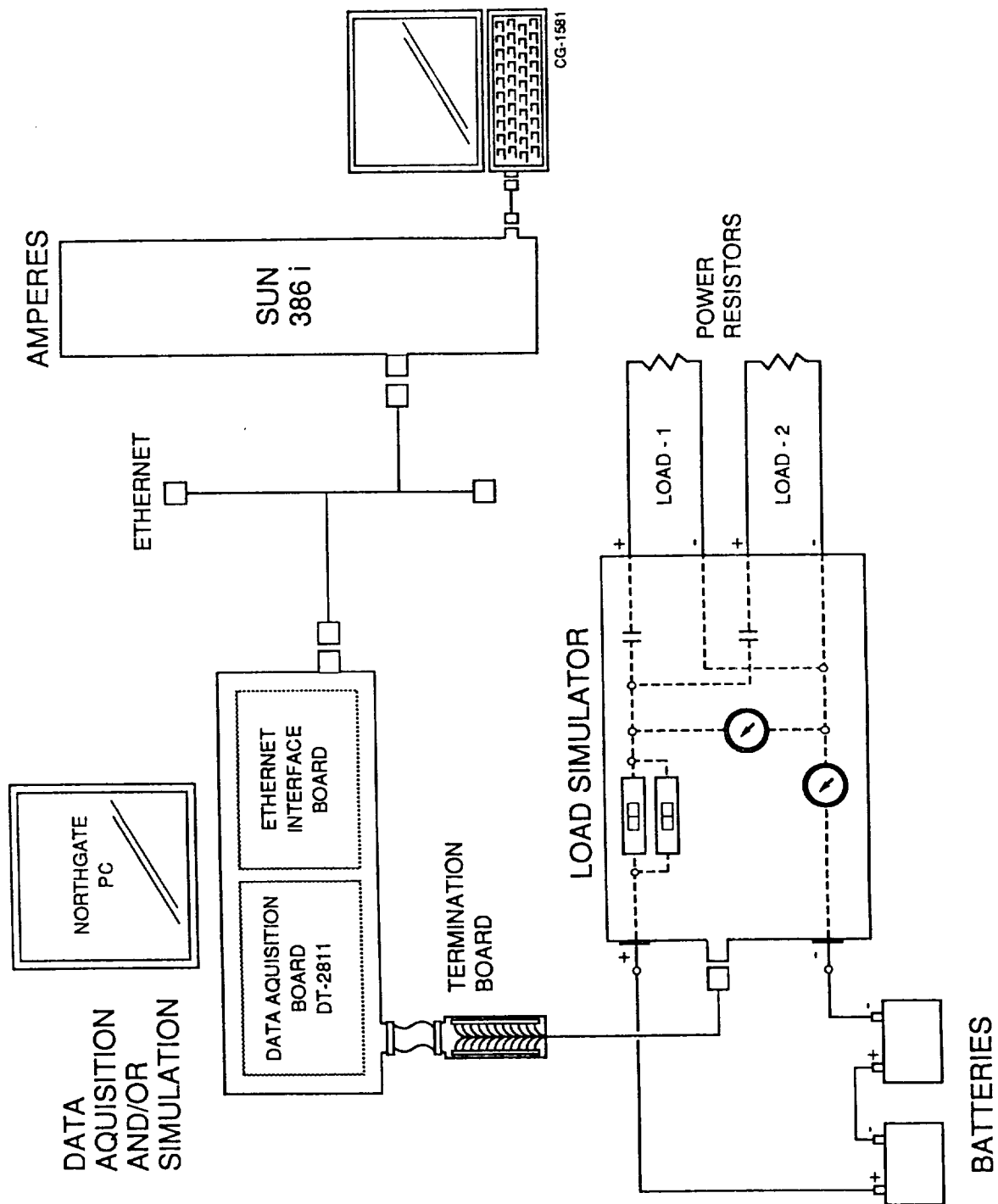
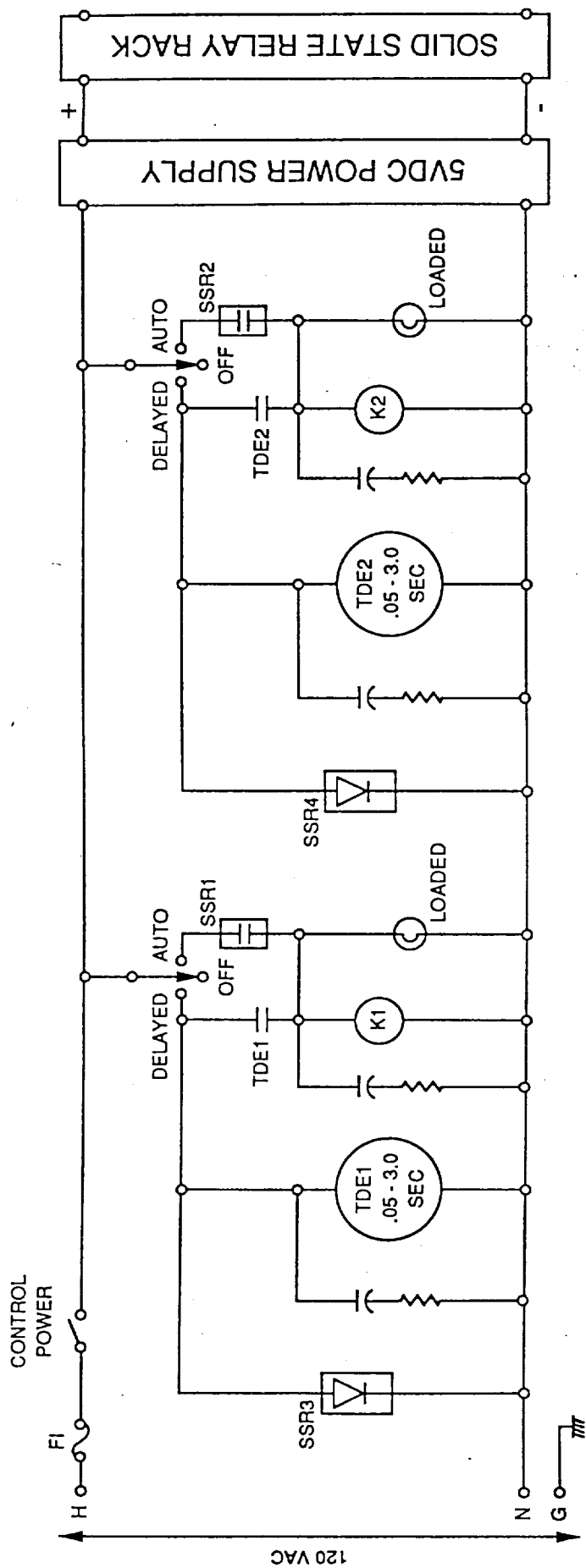


Figure 10. Data Acquisition System Block Diagram



NOTES

SSR1, 2 Solid State Output Relays

SSR3, 4 Solid State Input Relays
5, 6

TDE-1, 2 Time Delay Relays

CG-1580

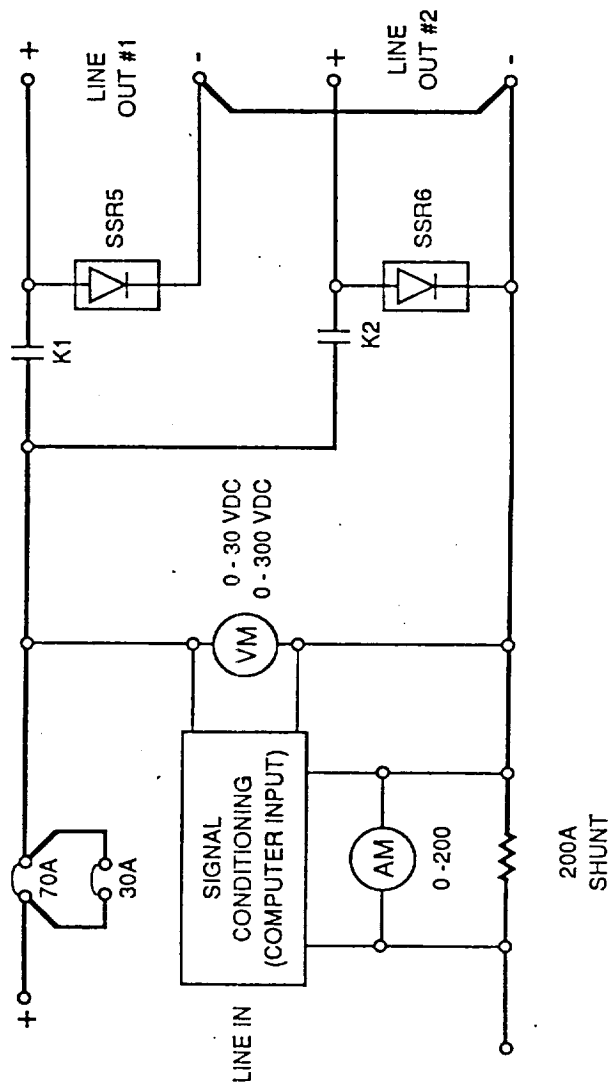


Figure 11. Fault Injection & Load Simulation Device

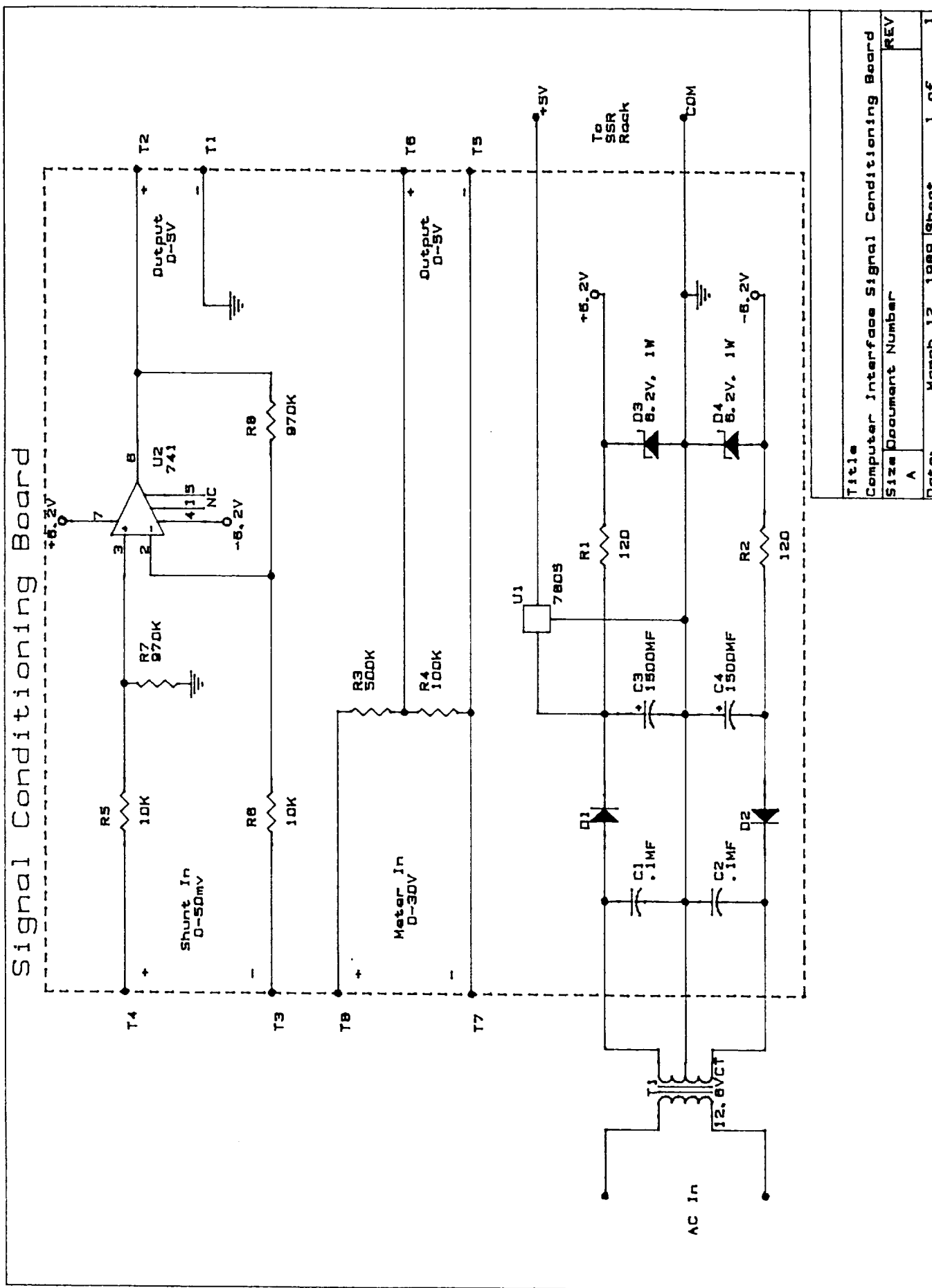


Figure 12. Signal Conditioning Board for the FILSD

4. Tasks Completed in Phase I

The following are the list of tasks completed in Phase I.

(1) Hardware selection and installation

i. Sun 386i was selected as main computer. Sun 386i was selected because of the following reasons:

a. Portability

It is a small size personal computer and easily portable.

b. Speed

It has a reasonably fast computational speed as a PC (5 MIPS).

c. Multi-tasking capability

It runs UNIX operating system enabling multi-tasking which is one of the major requirements in a real-time knowledge-based system.

d. User interface

It provides convenient program development environment and enables the development of a friendly user interface through the window tool kits.

Sun 386i is configured with 8 Mb of main memory, 3.5" 1.2 Mb floppy disk drive, 327 Mb of hard disk drive and 16", 1152 x 900 pixels color monitor.

Currently operating system takes 4 Mb of main memory and Sun Common Lisp takes another 4 Mb, and thrashing with swap space is frequently encountered. The main memory will be expanded to 12 Mb.

ii. Northgate 286 PC has been installed for load and fault simulation at the UTSI.

The detailed purpose of this PC is as follows:

a. Generation of the simulated operational data

Running a data simulation program, it generates the operational data based on various fault scenarios.

b. Data acquisition from the Fault Injection & Load Simulation Device (FILSD)

Actual current and voltage measurements are collected from the FILSD through the data acquisition board. The actual data thus collected can replace any one of the 10 simulated load data by the load simulation program.

c. Operational data transfer to the host computer through the Ethernet

The operational data, either generated by the simulation program or collected from the FILSD, is transferred to the host computer, Sun 386i, through the Ethernet.

d. Control of the FILSD

Load connection to the FILSD in load simulation mode or Fault injection into the AMPS with the FILSD in fault injection mode can be carried out by the PC either interactively or under program control.

- iii. Ethernet controller board, cables and necessary software on PC side were acquired and installed. Both loads and faults can be simulated using the FILSD and the data simulation program running on the Northgate PC and the simulated sensor data can be transferred to the sun 386i through the Ethernet for processing by the AMPERES.

(2) Software selection and installation

Originally IBUKI Lisp was selected and installed, since it was the only available lisp language on the Sun 386i as of Oct., 1988. The language was written in C, small in size but the language support was marginal. Sun Common Lisp was released to the UTSI for test in April, 1989. The test revealed that the Sun Common Lisp possesses various convenient features which are quite essential for real time knowledge-based system development, such as process forking off capability inside the lisp and sharing the same address space between the parent and the child process, process scheduling capability inside the lisp, and good documentation and language support, etc. Therefore the development language has been replaced from the IBUKI Lisp to the Sun Common Lisp. The speed of the Sun Common Lisp has not been confirmed yet because of the insufficient memory on Sun 386i, which will be expanded to 12 Mb in Phase II.

(3) Fault Injection & Load Simulation Device (FILSD) design and manufacturing

The FILSD was designed and assembled for load simulation and safe fault injection. Details of this device was explained in Section 2.3.

(4) Data acquisition system design and implementation

The data acquisition program written in C collects the data through the Ethernet and put them in circular buffers in a shared memory location such that AMPERES' fault monitoring and diagnosis process can access them. Details of the data acquisition system design and implementation was described in Section 2.2.

(5) Sensor value validation scheme development

One of the most crucial processes in real time fault monitoring and diagnosis is to validate various remote sensor values before using those values in any reasoning process. Trying to assess the system state based on the false sensor values is not only futile but may even be detrimental to the AMPS if any control action is taken based on the decision deduced from such false sensor values. A systematic sensor value validation procedure based on the logical environments and the casual relations of the sensors has been developed. Details of the sensor validation scheme was explained in Section 2.4.

(6) Representation scheme development for system components and configuration

System components are represented using the "structure" facility in Common Lisp. The system configuration information is embedded in the slot values of all the components such as "connect terminal." Details are explained in Section 2.5.

(7) Procedural knowledge representation scheme development

Procedural knowledge is represented with rule base. To reduce the total number of rules and to facilitate maintaining the consistency and integrity of the rule base, rules are expressed as generic as possible. Various rule languages are defined to express the rules close to the natural language. About 50 rules are defined in Phase I. Details are explained in Section 2.5.

(8) Main control and inference scheme development

Rules are grouped by their objectives to search efficiently and to facilitate maintaining consistency. Before invoking each rule group, the context is set for the execution of that specific rule group to simplify the rule syntax and to use an expression closer to natural language. Diagnosis procedure can be initiated from

any sensor measurement value showing the indication of the abnormality. It then examines the fault possibilities by looking at other sensor values with expectations. Once a diagnosis is made, further probe of the faults on different abstraction level can be pursued if necessary. Details are described in Section 2.6.

(9) System operational status and diagnosis result display

System operational status is displayed on the screen using windows. Four windows are used for display and operator input; the System Operating Status Window, the Major Operational Parameter Window, the Operator Interface Window, and the Fault Record Window.

The System Operating Status Window shows the one line diagram of the system and the current positions of all the circuit breakers and the magnetic switches. It also displays the major meter readings such as load currents, feeder currents, main current, main voltage, battery voltage, etc. If there exists any indication of an abnormality, sensors detecting that abnormality change the color to signify the findings.

The Major Operational Parameter Window displays major operational information of the power system and the AMPERES such as present time, main system voltage, main system current, present data buffer locations, orbital day or night, and overall system present operating state.

The Operator Interface Window displays the explanation about the current operating state and the results of the fault diagnosis. It also waits for the operator input.

The Fault Record Window displays the brief information for the past several faults. The operator can request detailed information about a specific fault by typing in the fault index number on the Operator Interface Window.

5. Project Schedule and Tasks Planned in Phase II

Overall project schedule is shown in Appendix i. Most of the Phase I work has been completed as of September 30, 1989. Demonstration of the prototype developed in Phase I will be scheduled with NASA/MSFC and Auburn Space Power Institute respectively.

In Phase II, operational real-time fault monitoring and diagnosis knowledge-based system integrated with the NASA test facility will be completed.

Details of the tasks planned in Phase II are as follows:

- (1) Battery short term and long term performance observation scheme design and implementation
 - i. Observation of the charge/discharge characteristics
 - ii. Early warning for the battery life
- (2) Solar array system performance observation scheme design and implementation
 - i. Observation of the I-V characteristics
 - ii. Observation of the Solar array output in relation with orbital locations
- (3) Operational data and fault data acquisition from the NASA test facility as the data becomes available and the operation of the AMPERES in real time
 - i. Application of the fault using the Fault Injection Device.
 - ii. Completion of data acquisition system including the installation of the standard communication protocols.
 - iii. Investigation of the system noise originated from switching surge, power source transition, etc.
- (4) Development of effective methods for monitoring dynamic loads
 - i. Handling of motor loads and the start-up inrush currents
 - ii. Handling of intermittent loads
 - iii. Handling of random loads
- (5) Power System reliability analysis
 - i. Collection of fault statistics
 - ii. Computation of LOLP (Loss of Load Probability)
- (6) Sensor value validation scheme development and reliability analysis

- i. Sensor value validation scheme using causal relations among sensors.
 - ii. Weibull and Weibayes analysis as sensor operational data accumulates.
- (7) Enhancement of the fault diagnosis capability by observing the short term trends of the sensor values
 - i. Implementation of the rule language “Observe”
 - ii. Association of the timer interrupt functions with the process created by the “Observer” function
- (8) Friendly user interface development
 - i. Detailed information display window (including orbital time)
 - ii. Display of faulty or live lines and components

6. Summary and Conclusion to Date

Many of the tasks involved in developing a real-time fault monitoring and diagnosis KBS, such as data structure for data acquisition, sensor value validation, reference operating state generation, effective inference scheme for fault diagnosis, etc., are still in research stages. Consequently carrying out this project requires both the research work and the implementation of the research results. Yet many of the concepts or approaches taken in this project should be refined and implementation details be elaborated.

The component centered approach is natural and effective since it follows the way how an experienced operator normally performs diagnosis. Necessary Meta knowledge should be formulated to decide whether the current diagnostic results offer sufficient information to the operator in proper abstraction level and to decide whether probing another abstraction level is necessary.

Sensor value validation procedure is required to develop a robust fault monitoring and diagnosis KBS working under the anticipated noises and disturbance caused by switching surges, and inductive load starts, etc. Short term, mid term and long term data observation is necessary for trend analysis and statistical analysis, which is essential for battery system diagnosis and will enhance the accuracy of the overall diagnosis results.

In Phase II, emphasis will be put on the completion of the research work initiated in Phase I and on the incorporation of the research results into the AMPERES to develop a practical real time KBS.

7. References

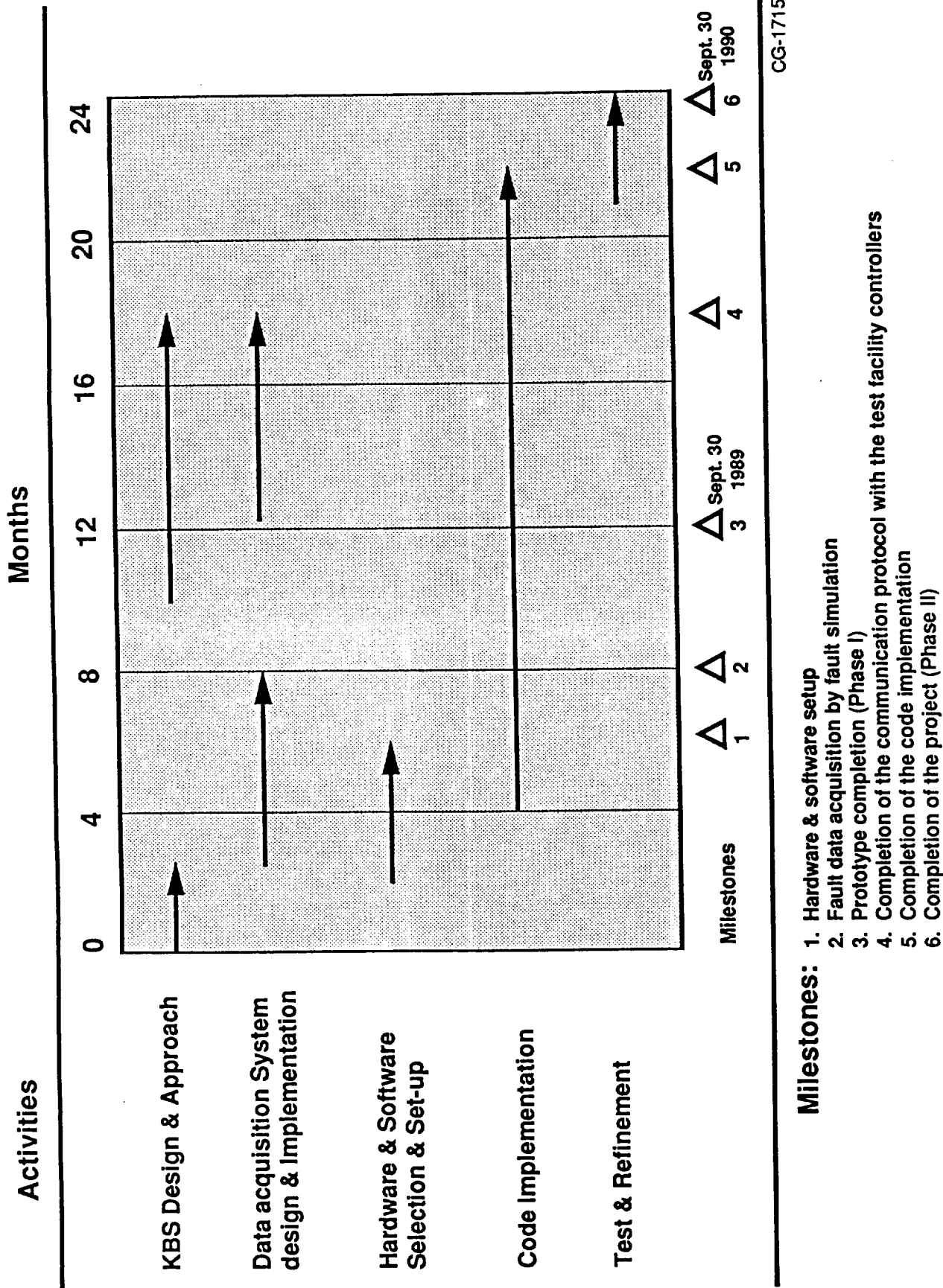
- [1] "Space Power Distribution System Technology: Autonomous Power Management," Final Report, Vol. I-V, TRW Report March, 1987.
- [2] D.J. Weeks, R.T. Bechtel, "Autonomously Managed High Power Systems," IECEC Conference Proceedings, 1985.
- [3] D.J. Weeks, "Space Power System Automation Approaches at the George C. Marshall Space Flight Center," IECEC Conference Proceedings, pp. 538-543, 1987.
- [4] Louis F. Lollar, "Automated Load Management for Space Craft Power Systems," IECEC Conference Proceedings, pp. 557-560, 1987.
- [5] B. Walls, "Expert System for Fault Detection and Recovery for a Space Based Power Management and Distribution System," IECEC Conference Proceedings, pp. 668-671, 1987.
- [6] S. Masui J. McDermott, A. Sobel, "Decision Making in Time Critical Situations," IJCAI-83, pp. 233-235, Los Altos, CA, 1983.
- [7] R.L. Moore, L.B. Hawkinson, C.G. Knickerbocker, L.M. Churchman, "A Real-Time Expert System for Process Control," 1st Conf. on AI Applications, pp. 569-576, Denver, CO, Dec., 1984.
- [8] M.E. Sorrells, "A Time-Constrained Interface Strategy for Real-Time Expert Systems," IEEE National Aerospace and Electronics Conference, pp. 1336-1341, Washington, D.C., 1985.
- [9] J. Marsh, J. Greenwood, "Real-Time AI: Software Architecture Issues," IEEE National Aerospace and Electronics Conference, pp. 67-77, Washington, D.C., 1986.
- [10] D. Leinweber, "Expert Systems in Space," IEEE Expert, Vol. 2, No. 1, pp. 26-36, 1987.
- [11] T.J. Laffey, P.A. Cox, J.L. Schmidt, S.M. Kao, J.Y. Read, "Real-Time Knowledge Based Systems," AI magazine Vol. 9, No. 1, pp. 27-45, Spring, 1988.
- [12] A. Wolfe, "An Easier Way to Build a Real-Time Expert System," Electronics, McGraw-Hill, March, 1987.
- [13] R. Davis, "Diagnosis via Causal Reasoning: Paths of Interaction and the Locality Principle," AAAI-83 Proceedings, Washington, D.C., 88-94, 1983.
- [14] R. Davis, "Diagnostic Reasoning Based on Structure and Behavior," Qualitative Reasoning about Physical Systems, Edited by D.G. Bobrow, The MIT Press, Cambridge, Massachusetts, 347-410, 1985.
- [15] T.L. Adams, "Model-Based Reasoning for Automated Fault Diagnosis and Recovery Planning in Space Power Systems," IECEC Conference Proceedings, pp. 1763-1769, 1986.

8. Appendix

(1) Project Schedule Attached

(2) Publications

- i. S.C. Lee, Louis F. Lollar, "Development of a Component Centered Fault Monitoring & Diagnosis Knowledge Based System for Space Power System," Proceedings of the IECEC-88, Denver, Colorado, Vol. 3, pp. 377-388, July 31 - Aug. 5, 1988.
- ii. L.D. Wilhite, S.C. Lee, L.F. Lollar, "Data Acquisition for a Real Time Fault Monitoring & Diagnosis Knowledge-Bases System for Space Power System," Proceedings of the IECEC-89, pp 117-121, Washington D.C., Aug. 6-11, 1989.
- iii. S.C. Lee, C. Patterson, M.W. Ratliff, F.W. Roepke, L.D. Wilhite, L.F. Lollar, "Real-time Fault Monitoring & Diagnosis Knowledge-based System for Space Power Systems: AMPERES," will be published soon.
- iv. S.C. Lee, "Sensor Value Validation Based on Causal Relations," will be published soon.



CG-1715

- Milestones:**
1. Hardware & software setup
 2. Fault data acquisition by fault simulation
 3. Prototype completion (Phase I)
 4. Completion of the communication protocol with the test facility controllers
 5. Completion of the code implementation
 6. Completion of the project (Phase II)

Figure 3. Project Schedule